Template Models

Overview

Sharing of both structure and the parameters across different template variables.

- **Template variable** $X(U_1...U_k)$ is instantiated multiple times. For example, the Grade variable for different class
- **Template Model** is languages that specify how (ground) variables inherit dependency model from template
- Dynamic Bayesian networks temporal data
- Object-relational models, like people, courses

Temporal Models

A system evolved over time.

Distributions over Trajectories

Since continued parameters are hard to deal with, we usually discretize time continued variable

- Pick time granularity Δ
- $X^{(t)}$ variable X at time $t\Delta$
- $X^{(t:t')} = \{X^{(t)}...X^{(t')}\}, (t \leq t')$

- Want to represent $P(X^{(t:t')})$ for any t and t'

Markov Assumption

For temporal case, we can draw joint distribution as $P(X^{(0:T)}) = P(X^{(0)}) \prod_{i=0}^{T-1} P(X^{(t+1)}|X^{(0:t)})$. This formula has no assumption. To calculate this we need to know all the state from 0 to $t-1$, which is computation-consuming.

Markov Assumption

Then we assume that once we know the present, we forget about past, and next state is only dependents on the current. Formally, $(X^{(t+1)} \perp X^{(0:t-1)}|X^t)$

Apply this assumption to the above equation, we have $P(X^{(0:T)}) = P(X^{(0)}) \prod_{i=0}^{T-1} P(X^{(t+1)}|X^{(t)})$.

Sometimes this assumption is too strong to the case, in order to fix it, we usually have two main methods:

- Enrich the state description

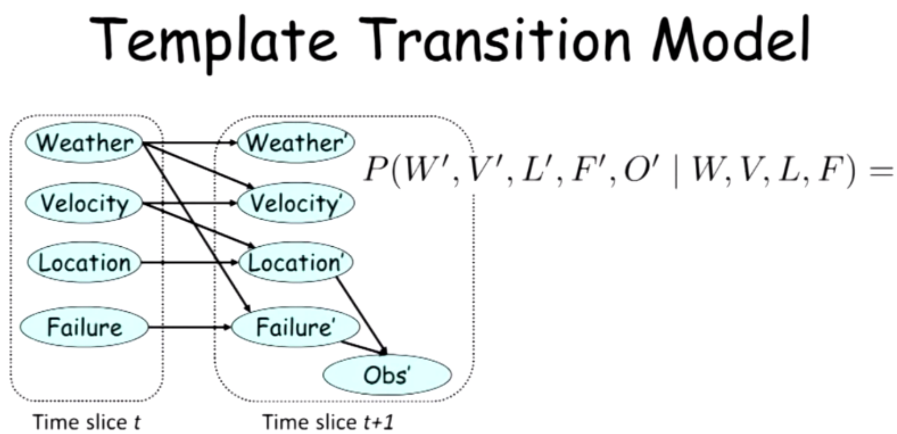
- Go further back in time, not just one time slice. This also called semi-Markov

Time Invariance

From Markov assumption, we can get a model template that assume model is duplicated for every single time point. $P(X'|X)$, where X' is the next time state and X is current state. The model won't be change along time.

Then for all t: $P(X^{(t+1)}|X^{(t)}) = P(X'|X)$

Template Transition Model



This graph describe a conditional distribution of time t given $t-1$. There are two types of edges in this graph:

- The edge across the time slice. This called **inter-time-slice**. And the edge from one variable to next state of it is called **persistence edge**, like the edge from Velocity to Velocity' in graph.
- Edge from RV to RV in same time slice. This called **intra-time-slice**. Which often refers to the events that happens rapidly.

2-time-slice Bayesian Network

A transition model(2TBN) over $X_1 ... X_n$ is specified as a BN fragment such that:

- The nodes include $X'_1 ... X'_n$ and a subset of $X_1 ... X_n$. Its subset since it only store the RVs that can directly affect the next state.
- Only the nodes $X'_1 ... X'_n$ have parents and a PCD

The 2TBN defines a conditional distribution $P(X'|X) = \prod_{i=1}^{n} P(X'_i|Pa_{X'_i})$

Dynamic Bayesian Network

A dynamic Bayesian network (DBN) over $X_1...X_n$ is defined by a

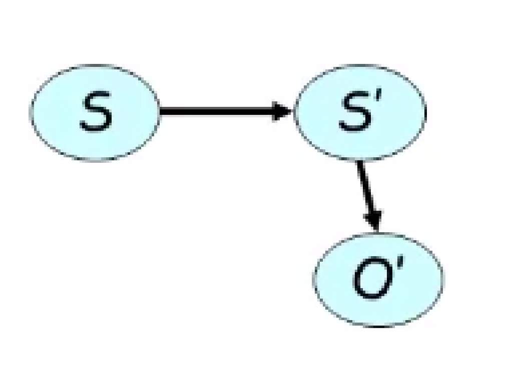
- 2TBN $BN_{\rightarrow}$ over $X'_1...X'_n$
- a Bayesian network $BN^{(0)}$ over $X_1^{(0)}...X_n^{(0)}$. This is initial state

For a trajectory over $0, ..., T$ we define a ground (unrolled network) such that

- The dependency model for $X_1^{(0)}...X_n^{(0)}$ is copied from $BN^{(0)}$.
- The dependency model for $X_1^{(t)}...X_n^{(t)}$ for all \$t >0 \$ is copied from $BN_{\rightarrow}$

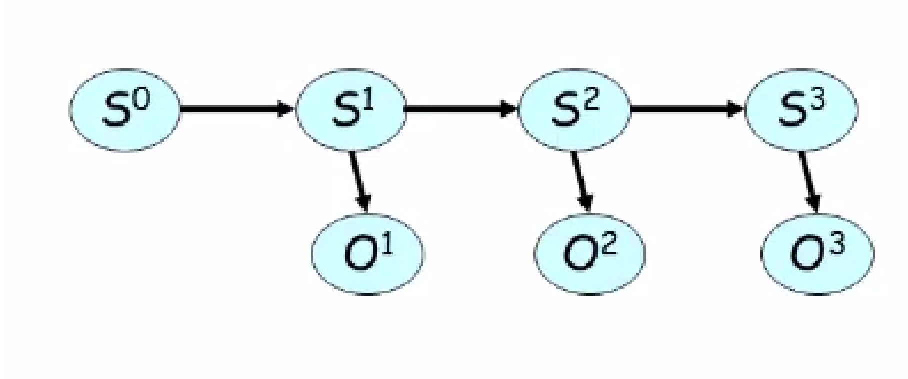
Hidden Markov Model

Hidden Markov Model can be simply viewed as probabilistic model that has a state variable S and a observation variable O. As following graph:



Therefore, we have a transition model that tell us how states change to next state and a observation model that can get observation we want from specific state.

Then we can unroll above 2TBN. We can get structure like following graph. It should be noticed that S_1 here is not random variable. It is assignment of S. The assignment of S might be change over time.



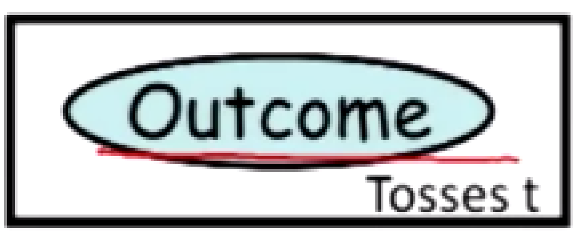
Summary

- HMMs can be viewed as a subclass of DBNs
- HMMs seems unstructured at the level of random variables.
- HMM structure typically manifests in sparsity and repeated elements within the transition matrix

Plate Models

Modeling Repetition

Image a case we toss the coin. We toss coins k times and we will get k outcomes. If we use RV to represent these outcomes, we will have k RVs in our graph. In order to avoid the repetition of this process, we can use template. In graph, we usually use a box to round the RV. As shown in figure.



Then, we can say the outcome variables is indexed by the toss t. This template can represent a set of random variables that represent k coin tosses.

One of benefits that using plate model is that we can make the parameters independent from the plate. Which means the RVs that from same plate can use same PCD parameterization.

Nested Plates

Now image the student example again, we have courses c and students s. The difficulty is dependents on specific course, every course has its own difficulty. And every student has Intelligent and Grade.

As mentioned, we can come out with two plates, first one has only difficulty variable, which indexed by course c. Second one has two variables, Intelligence and Grade, and its indexed by student s. However, we want that each student has different intelligence and therefore has different grade for different courses. Then we can use **Nested Plates** as shown in figure:

Note: Intelligence and Grade in graph are indexed by both s and c

Overlapping Plates

Now we want the Intelligence of students has no relationship with c. We can use **Overlapping Plates** to represent:

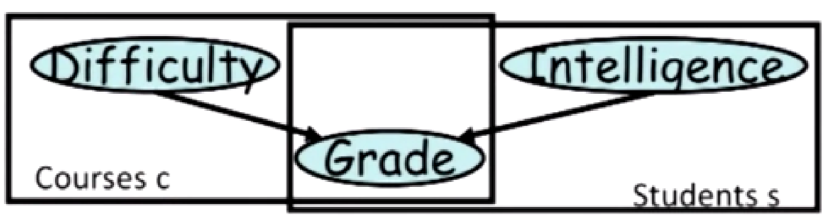


Plate Dependency Model

- For a template variable $A(U_1...U_k)$:
  - Template parents $B_1(U_1)...B_,(U_m)$, we doesn't have an index in the parent that doesn't appear in the child. Which means the index for parents are subset of index of child.
  - CPD $P(A|B_1...B_m)$. This template model can represent a variable that have unbounded number of parents.

Summary

- Template for an infinite set of BNs, each induced by a different set of domain objects.
- Parameters and structure are **reused** within a BN and across different BNs.
- Models encode correlations across multiple objects, allowing collective inference.
- Multiple “languages”, each with different tradeoffs in expressive power.